are passed back from the data base descriptor manager for conversion to textual form and insertion into the data base catalog (line 213).

Constraint Enforcement Using the Relationship Descriptors

When an INSERT or LOAD operation is made on a table, that table's dependent chain of relationship descriptors is traced and the constraints so located are enforced. First, the table's record descriptor is located. If field REC.B in the record descriptor is zero (0), then the dependent chain is empty and no further action is required.

If the record descriptor's REC.B field is nonzero, then the chain of relationship descriptors which originates there is traced and processed. For each relationship descriptor so located, the description of the foreign key in field REL.F is used to construct the foreign key contained in the record being loaded or inserted. If the resulting foreign key value is not null, it is used as a search argument against the primary key index to verify the existence of a row in the parent table with the matching primary key value. If no such row is found, the referential constraint of that relationship descriptor would be violated by the new row to be inserted or loaded, and the operation is disallowed. However, if all relationship descriptors in the dependent chain are satisfied, the INSERT or LOAD is allowed to proceed.

For an update of a table's primary key field or fields, the parent chain of relationship descriptors anchored in the REC.A field of the table's record descriptor is traced and processed. Each relationship descriptor located on the parent chain is used to determine the UPDATE rule to apply, to determine the dependent table, and to locate any dependent rows with a foreign key that matches the primary key being updated. The UPDATE operation is allowed or disallowed by enforcing the UPDATE rules so located, substantially as described above under the heading "Referential Integrity". If the UPDATE rule is "CASCADE", the parent chain of each dependent table so located is also traced and processed as described.

If a foreign key field is updated, the dependent chain of relationship descriptors anchored in REC.B is traced and processed. Each relationship descriptor that describes a foreign key that contains the modified field(s) is used to construct the updated foreign key value. If the updated value is not null, it is then used as a search argument against the primary key index identified in the relationship descriptor to verify the existence of a parent row with the primary key value matching the updated foreign key value. The UPDATE operation is disallowed if no such row is found for any one of the relationship descriptors on the dependent chain.

When a row is deleted, the parent chain of relationship descriptors anchored in REC.A of the deleted row's record descriptor is traced and processed. Each relationship descriptor located on the parent chain is used to determine the DELETE rule to apply, to identify the dependent table, and to locate any dependent rows with foreign key values matching the primary key value of the row being deleted. The DELETE operation is either carried out or disallowed depending on the particular DELETE rules stored in the REL.K fields of the relationship descriptors, again substantially as described above under the heading "Referential Integrity". If the DELETE rule is "CASCADE", the parent

chain of each dependent table so located is also traced and processed as described.

FIG. 7 shows the relational database management system in block diagram form. The system includes at least two relational tables **90**, at least one relationship descriptor **91** describing a referential constraint between the relational tables, record descriptors **92** providing access paths between the relational tables and the relationship descriptor, access means **93** for accessing the relationship descriptor through symbolic pointers when a relational table is to be modified, and enforcement means **94** for enforcing the referential constraint described by the relationship descriptor upon modification of a relational table.

It will be appreciated that, although a specific embodiment of the invention has been described herein for purposes of illustration, various modifications may be made without departing from the spirit and scope of the invention. For example, offsets, direct pointers or symbolic locations could be substituted for the symbolic pointers used to access the record, index, and relationship, descriptors. Accordingly, the scope of protection of this invention is limited only by the following claims.

We claim:

1. A computer-implemented, relational data base management system comprising:

(a) a first table containing data records;

(b) a second table containing data records;

(c) a third table containing data records;

(d) a first relationship descriptor, having pointer connections to the first table and the second table, for describing meta-data of a first referential constraint between the first and second tables, the meta-data including the first constraint's parent and dependent tables and primary and foreign keys, the first relationship descriptor being a separate object internal to the data base management system;

(e) a second relationship descriptor, having pointer connections to one of the first two tables and the third table, for describing meta-data of a second referential constraint between one of the first two tables and the third table, the meta-data including the second constraint's parent and dependent tables and primary and foreign keys, the second relationship descriptor being a separate object internal to the data base management system;

(f) a first chain for connecting the first and second relationship descriptors to their respective parent tables;

(g) a second chain for connecting the first and second relationship descriptors to their respective dependent tables;

(h) means for accessing the first and second relationship descriptors when any of the tables is to be modified; and

(i) means for enforcing the referential constraints described by the first and second relationship descriptors upon such modification.

2. The system of claim **1**, wherein the first relationship descriptor is stored within the data base system in compiled form.

3. The system of claim **1**, further comprising:

a first record descriptor for describing the data records of the first table, the first record descriptor being a separate object within the data base system which can be modified individually; and

wherein the means for accessing the first relationship descriptor comprises a logical connection between